Throttling Twitter: An Emerging Censorship Technique in Russia

Diwen Xue
University of Michigan

Reethika Ramesh
University of Michigan

ValdikSS
Independent

Leonid Evdokimov
Independent

Andrey Viktorov
Independent

Arham Jain
University of Michigan

Eric Wustrow
University of Colorado Boulder

Simone Basso
OONI

Roya Ensafi
University of Michigan

ABSTRACT

In March 2021, the Russian government started to throttle Twitter on a national level, marking the first ever use of large-scale, targeted throttling for censorship purposes. The slowdown was intended to pressure Twitter to comply with content removal requests from the Russian government.

In this paper, we take a first look at this emerging censorship technique. We work with local activists in Russia to detect and measure the throttling and reverse engineer the throttler from in-country vantage points. We find that the throttling is triggered by Twitter domains in the TLS SNI extension, and the throttling limits both upstream and downstream traffic to a value between 130 kbps and 150 kbps by dropping packets that exceed this rate. We also find that the throttling devices appear to be located close to end-users, and that the throttling behaviors are consistent across different ISPs suggesting that they are centrally coordinated. Notably, this deployment marks a departure from Russia's previously decentralized model to a more centralized one that gives significant power to the authority to impose desired restrictions unilaterally. Russia's throttling of Twitter serves as a wake-up call to censorship researchers, and we hope to encourage future work in detecting and circumventing this emerging censorship technique.

CCS CONCEPTS

• **General and reference** → **Measurement**; • **Security and privacy** → **Security protocols**; • **Social and professional topics** → **Governmental surveillance**; *Technology and censorship.*

KEYWORDS

Censorship, Throttling, Interception, Russia

ACM Reference Format:
Diwen Xue, Reethika Ramesh, ValdikSS, Leonid Evdokimov, Andrey Viktorov, Arham Jain, Eric Wustrow, Simone Basso, and Roya Ensafi. 2021. Throttling Twitter: An Emerging Censorship Technique in Russia. In *ACM Internet Measurement Conference (IMC '21), November 2–4, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3487552.3487858

IMC '21, November 2–4, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-9129-0/21/11.
https://doi.org/10.1145/3487552.3487858

1 INTRODUCTION

Traditional Internet censorship relies on targeted blocking of content and resources. Censors implement blocking using network traffic features such as IP [8, 12, 13], DNS [6, 16, 30, 37], keywords [15, 36, 57], or protocol fingerprints [3, 7, 11, 56]. In extreme cases, censors have also used Internet shutdowns to completely sever connection to the Internet to prevent unwanted access [21, 22, 49].

While blocking is a common tool for censors, less has been observed about *throttling* connections as a means for censorship. In contrast to blocking, throttling aims to degrade bandwidth to a service to discourage its use while still allowing some access. This offers an attractive technique for censors as it is more difficult for users and circumventors to detect or attribute the slowdown to censorship: slow connections may be a natural result of network congestion and not intentional throttling.

In March 2021 the Russian government started throttling Twitter on a national scale [34], in an attempt to pressure Twitter to comply with Russian content removal requests [42]. While throttling an entire user Internet connection near political events has been observed before such as in Iran in 2013 [55], Russia's slowdown of Twitter marks the first instance of a country *selectively* throttling specific domains and services on demand as an emerging new censorship technique. Under pressure, Twitter fulfilled the majority of content takedown requests to comply with the Russian government's order without providing any transparency to its users. In May 2021 Russia threatened to use the same throttling technique against Google in response to disputes over anti-government content on YouTube [40].

In this paper, we investigate and document the Russian throttling of Twitter in depth. Hours after the onset of the throttling we started conducting measurements using multiple in-country vantage points to investigate the behavior of the throttling and how it changes over time. In addition, we use public crowdsourced data covering 401 unique Russian ASes to measure how widely the throttling impacted Internet users in Russia.

Our findings show that the throttling is triggered upon observing Twitter-related domains (`*.twimg.com`, `twitter.com`, `t.co`) in the SNI (Server Name Indication) extension of a TLS Client Hello record. The throttling is not symmetric and can only be triggered for TCP connections that originate from within Russia. However, once such a connection is established, throttling can be triggered by a Twitter SNI sent in either direction. Moreover, we observe that the throttling devices inspect beyond the first packet in a connection

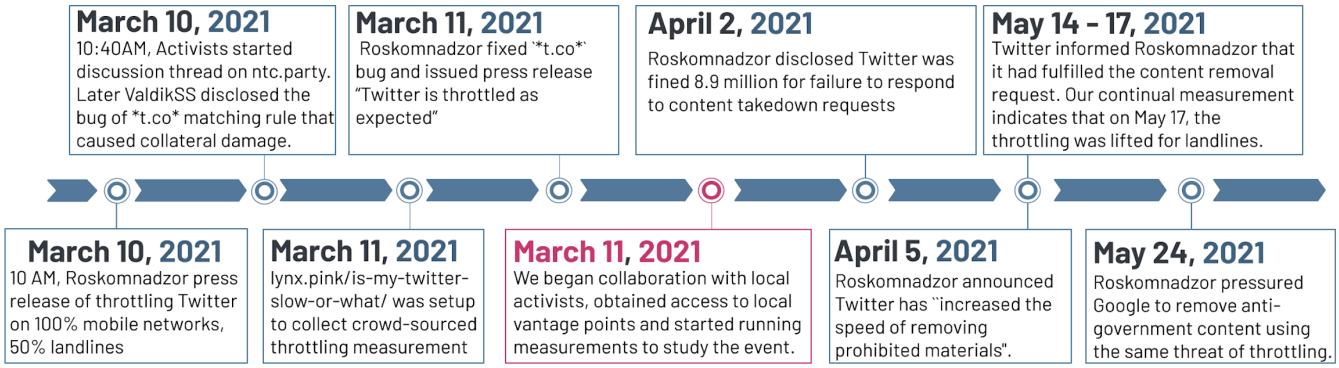

Figure 1: Timeline of the Twitter throttling incident.

(where typically the SNI-containing Client Hello message would appear), possibly as a countermeasure to circumvention attempts. Once the throttler is triggered, data packets transferred in either direction (download/upload) will be dropped once the rate limit (around 130 kbps to 150 kbps) is reached. We also perform TTL-limited measurements and determine that the throttling devices are placed close to end-users but are not co-located with the ISP devices performing blocking, suggesting they may be separate from existing blocking infrastructure. Finally, we find that the throttling behaviors are largely consistent across different ISPs, suggesting that the throttling devices are likely centrally coordinated.

Based on our measurements of the throttling mechanism, we make several recommendations on how to circumvent the throttling, such as TCP-level fragmentation or TLS packet stuffing. We also recommend that browsers and websites implement efforts to support TLS Encrypted Client Hello (ECH) to make it more difficult for censors to throttle based on SNI.

To the best of our knowledge, our work is the first to study and analyze targeted throttling at a national scale. We anticipate that governments' next-generation censorship techniques will target degrading quality of service of sensitive domains in similar ways, making this an important problem to study, especially since current censorship detection platforms [33, 35, 50] focus on blocking and are not yet equipped to monitor throttling. We hope our work encourages future work in detecting and circumventing this emerging censorship technique.

2 BACKGROUND

Traffic throttling: Throttling is an intentional act by an ISP or other network intermediary to reduce the bandwidth allocated to network traffic. There are two common ways to implement throttling: *traffic shaping*, which *delays* packets exceeding an assigned rate limit, and *traffic policing*, which *drops* the exceeding packets instead [9]. Throttling can be either targeted, applied to only a select set of protocols or users, or indiscriminate, applied to all traffic regardless.

There is a limited literature that studied traffic throttling. Kakhki et al. designed an app that uses a "Record-and-Replay" method to detect throttling for arbitrary applications on mobile networks [23]. Flach et al. developed heuristics to quantify traffic policing from server-side traces [17]. Furthermore, Li et al. developed a methodology and a tool to identify traffic classification rules that trigger throttling from middleboxes [26, 28]. Their analysis revealed 30 ISPs in 7 countries that deployed traffic throttling mechanisms [27].

While there are instances of ISPs offering different performance for different users or services [10, 27, 59], throttling used for censorship is largely unprecedented. One exception is the nationwide Internet slowdowns in Iran during periods of political upheavals [4, 14, 55], but those events were not targeted but instead applied to all traffic. In this paper, we investigate the first ever use of large-scale, targeted throttling for censorship purposes.

Changes in Russia's Censorship Model: Russia's network architecture consists of thousands of ASes and a large number of ISPs, which is similar to many other countries around the world. As shown by Ramesh et al., unlike China and Iran, Russia uses a decentralized information control mechanism with different ISPs implement censorship differently, hence contributing to the fragmentation of access to online content for users in Russia [39].

Specifically Ramesh et al. showed that each ISP is responsible for the DPI (deep packet inspection) systems under their control. Roskomnadzor (Russia's authority on information control) provides a list of blocked resources, over 100k domains and IPs, to be downloaded and used by each ISP's DPI system. Many ISPs use commercially available hardware solutions, but some used open source filtering software or implemented their their own.

However, as we show in this paper, the behaviors of the throttlers show a high degree of coordination across different ISPs. This marks a departure from the decentralized model, which suggests that Roskomnadzor is successfully moving towards centralized control on its decentralized network of thousands of ISPs.

3 ETHICS

Measuring censorship events raises important ethical considerations that require due diligence from researchers to protect any human subjects involved. Most such studies, including ours, measure censorship policies by actively triggering the censors and observing their responses, which may put participants at risk. We carefully designed our measurements to follow best practices described in Menlo [32] reports and we were guided by several ethical considerations from previous works [39, 52].

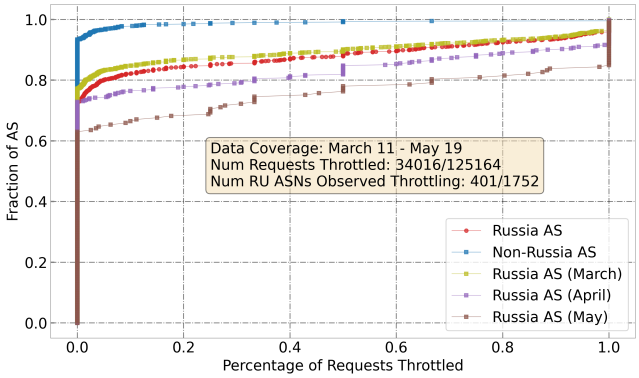

Figure 2: Fraction of requests throttled at Russian / non-Russian AS level

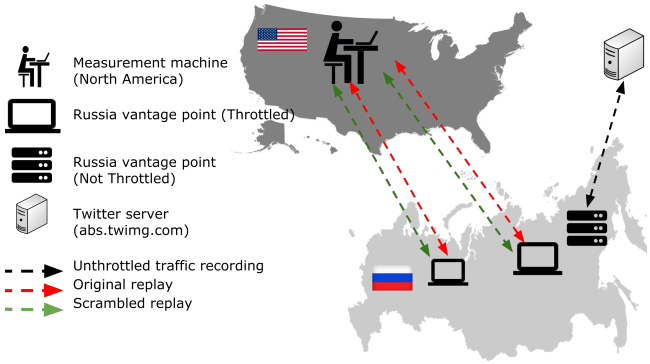

Figure 3: Record and Replay measurement setup

We use two primary data sources: 1) measurements conducted from our own in-country vantage points in Russia due to asymmetric nature of throttling (see § 6.5) and 2) crowd-sourced dataset of the throttling. Before performing any measurements, we carefully discuss the details of our tests and only proceed after getting consent from the owner of the vantage point. We are not aware of anyone who has been arrested or fined by the government for performing this type of measurement, and we ourselves have performed this kind of research in prior work and are aware of the risks.

With respect to the usage of the public, crowd-sourced measurement dataset (see § 4), we reached out to our US institution's IRB and we obtained an official determination from the IRB as *Not Regulated*. Nonetheless, we make sure that the data was collected ethically. The website measures and compares client's bandwidth to Twitter and to a control site by sending requests and timing the downloads and was set up by one of the authors. Note that accessing Twitter was permitted in Russia even while it was throttled. In addition, before starting any measurement, this website informs the users about the description of the tests, the data collected, and links to open-sourced code. It also explicitly states that it collects timestamps, speeds for each test case, IP (anonymized to subnet), Autonomous System Number, and ISP information. All data was bucketed into 5-min bins before being made public in order to eliminate any time correlation.

4 THROTTLING TWITTER INCIDENT

On March 10 2021, Roskomnadzor announced that the government had "taken measures to protect Russian citizens from the influence of unlawful content" and began throttling Twitter due to its non-compliance with Russian content takedown requests [42]. According to the statement, the throttling of Twitter was implemented on 100% of mobile services and 50% of landline services. On April 5 2021, Roskomnadzor gave an ultimatum to Twitter to fulfill their requirements by May 15 to avoid being completely blocked [44]. Under pressure, Twitter removed 91% of the requested prohibited content and, as a result, throttling was lifted on landlines on May 17, while it remains throttled on mobile services [45, 46] at the time of submission. Figure 1 shows a timeline of the throttling incident. A more extensive record can be found in Appendix A.1.

The first reports about the throttling came from Russian activists on `ntc.party`, a forum for network censorship. Consequently, a website was set up to collect crowd-sourced measurements from users by fetching an image hosted on Twitter and non-Twitter domains and comparing the performance [53]. The dataset is publicly available at [5].

Analyzing this data, we find that the throttling of Twitter in Russia is widespread. Figure 2 shows fractions of requests throttled at the AS level. From March 11 to May 19, the website recorded 34,016 measurements from 401 unique Russian ASes that show large slowdowns in speeds for the Twitter requests.

Anecdotal reports from the Russian Internet freedom community suggest that the throttling is being implemented with so-called *TSPU* (технические средства противодействия угрозам technical solution for threat countermeasures). As later confirmed by a government official, TSPU is a deep packet inspection (DPI) boxes specifically developed by RDP.RU on Roskomnadzor's orders [1, 51]. Unlike existing middleboxes used for filtering by individual ISPs, TSPU devices are "with in the framework of centralized control", i.e. they are directly controlled by Roskomnadzor [43].

While the incident gained public and media attention, the implementation details and devices behind the throttling still remained a blackbox for the community. Several questions remained unanswered: How is the throttling implemented? Where in the network does the throttling occur? What exactly triggers the throttling? How can the throttling be circumvented? Is the throttling stable/consistent over time?

5 MEASUREMENT SETUP

Working extensively with the local Internet freedom community, we secured eight local vantage points as listed in Table 1. By comparing the available bandwidth to Twitter domains with random, non-Twitter domains, we established that seven of them experienced throttling at the time. The un-throttled vantage point served as control for our measurements. Next, we set up our vantage points to follow the "record and replay" approach introduced by Kakhki et al. [23] to reverse engineer how the throttler works. This technique works by recording an un-throttled connection and using a vantage point in the tested network to replay the recorded transcript in order to infer if throttling is triggered in that network. The replay system imposes a few restrictions intended to capture

Mobile ISP	Throttled as of 3/11?	Landline ISP	Throttled as of 3/11?
Beeline	Yes	OBIT	Yes
MTS	Yes	JSC Ufanet	Yes
Tele2	Yes	JSC Ufanet	Yes
Megafon	Yes	Rostelecom	No

Table 1: Vantage points in Russia used in our study

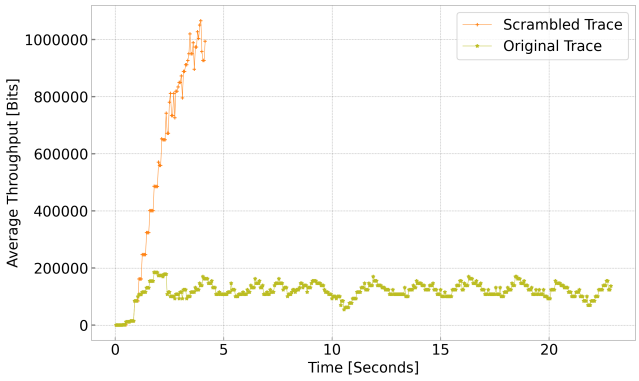

Figure 4: Original and Scrambled replay throughput

the nuances of the recording (such as inter-packet logic) but leaves all other aspects to the TCP stack of each endpoint. Essentially, traffic being replayed is identical to the recorded session, except for the server IP address which has changed from Twitter to the replay server. We highlight that the replay system does not perform any DNS lookup, nor does it communicate with the actual Twitter server in any way. The goal of the replay system is to detect any content-based traffic differentiation policy along the network path between the client and the replay server.

Figure 3 illustrates this setup. First, we collect a trace using packet captures on the unthrottled vantage point while fetching an 383 KB image from `abs.twimg.com`. We also record a trace where upload traffic dominates the bandwidth by uploading the same image to a server under our control, preceded by a Twitter Client Hello. Then, we set up a replay server at our university and use all our Russian vantage points as replay clients.

To establish a baseline for the throttling, for each vantage point, we first replay the original Twitter traffic recording, which triggers throttling (Original Trace in Figure 4). Next, we replay the same recording but with each TCP payload byte inverted so that any structure or keyword that may trigger the throttling is removed (Scrambled Trace in Figure 4). The choice of using bit-inverted replays as control was inspired by previous works [27, 28], which found such technique was able to successfully evade DPI detection. With multiple experiments across different vantage points, we found that **the throttling throughput converges to a value between 130 kbps and 150 kbps** for both the download and the upload replays.

6 REVERSE ENGINEERING THE THROTTLER

Upon confirming the presence of throttling, we conduct in-depth measurements to understand the nuances of the throttling and to reverse engineer the way it works. We note that, unless otherwise

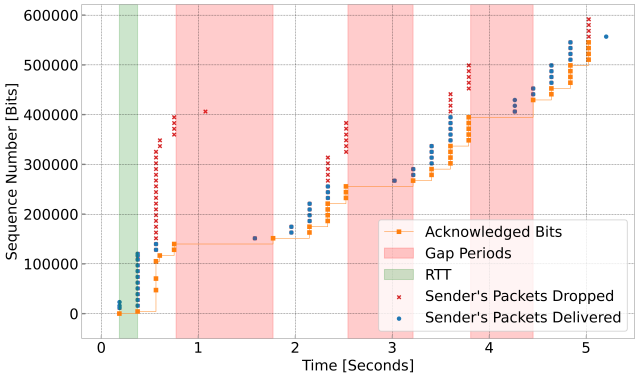

Figure 5: Sequence numbers as seen by sender and receiver. "Gaps" correspond to intervals during which no packet is delivered to the receiver

stated, the same measurement results were obtained from all vantage points experiencing throttling. This high degree of uniformity in our measurement results across different ISPs suggests that these throttling devices might be centrally coordinated.

6.1 Throttling Mechanism

We compare server-side and client-side packet captures (pcaps) of throttled replay experiments to understand how the throttling is implemented.

The throttler uses traffic policing: We find that the throttling is implemented by dropping packets that exceed a rate limit. Figure 5 shows the sequence number evolution as seen on one Russian client and the university server sending data. Comparing the sequence numbers of the packets sent by the sender (red and blue dots) with the ones delivered to the client (blue dots only), we find that packets exceeding a certain rate limit are silently dropped in transmission, resulting in "gaps" over five times the typical RTT.

On some cellular vantage points, we observe other throttling policies in addition to the throttling of Twitter. For instance, on Tele2-3G, *all* our upload traffic is slowed down using delay-based shaping as illustrated by the smoothed curve in Figure 6, as opposed to the saw-tooth shape that corresponds to loss-based policing. Note that this observed upload bandwidth of 130kbps is not specific to Twitter, as all traffic are being slowed down regardless of SNI or destination. On the other hand, most download traffic is not affected, except for Twitter traffic which triggers throttling behaviors similar to what we observed from other ISPs.

When multiple throttling schemes are being used by different network intermediaries at different network locations (e.g. the upload slowdown on Tele2-3G could be due to the subscribed asymmetrical Internet plan), pinpointing the reason and isolating one specific throttling scheme from others can be difficult. In our case, even though we ran multiple replay experiments on Tele2-3G, we were not able to conclude whether a specific upload throttling policy exists for Twitter, because all upload traffic was being throttled at a slightly lower rate. We, therefore, exclude Tele2-3G when analyzing upload measurements.

6.2 Triggering the Throttling

To identify which packets of a connection and what parts of those packets trigger throttling, we craft several different initial packet sequences to send to the server and monitor the throttling.

The throttler parses network packets from both directions and throttles the connection upon observing a sensitive TLS SNI in a Client Hello: We start by testing if a sensitive SNI in Client Hello record *alone* is enough to trigger throttling. To test this, we replay a traffic capture with a Twitter domain in the TLS SNI between a Russian client and our server outside the country. Next, we randomize all packets of the same traffic capture except the Client Hello. In both cases, we observe throttling on the connection, suggesting that a sensitive Client Hello is sufficient to trigger throttling. Furthermore, we find that a Client Hello with a Twitter SNI sent by the replay server also triggers the throttling, suggesting that the throttler inspects both upstream and downstream traffic. We investigate the symmetry of throttling in § 6.5.

We conduct measurements to understand if the throttler ever stops looking for a trigger. We prepend a packet with random bytes of varying sizes before the triggering Client Hello. For all the trials with the random packet size over 100 bytes, we did not observe any throttling. This suggests that the throttler, upon seeing a packet that cannot be parsed into any protocol it supports, will stop inspecting the packets that follow, likely in an effort to conserve the DPI's resources. However, if we send any valid TLS record, HTTP proxy packet, SOCKS proxy packet or a random packet with less than 100 bytes, the throttler continues to inspect for an additional 3-15 packets in the session. This behavior may be designed to target circumvention techniques that work by inserting a fake Client Hello (e.g. GoodbyeDPI [19], Zapret [58]) or by routing traffic through unencrypted proxies.

Focusing on the triggering Client Hello, we follow a binary search approach introduced by previous works [26, 28] to identify which parts of the packet are inspected by the throttler. We do this by recursively masking (with inverted bits) half of the Client Hello payloads in order to identify which bytes or fields within the Client Hello trigger throttling. We find that if we mask fields such as *TLS_Content_Type*, *Handshake_Type*, *Server_Name_Extension*, or *Servername_Type*, the connection does not trigger throttling. This suggests that the throttler inspects only certain TLS packets (e.g. Client Hello containing SNI) and parses the packet for the SNI, rather than simply regex-matching the Twitter domain string over the entire packet. Moreover, tampering with *TCP_Length*, *TLS_Record_Length*, or *Handshake_Length* thwarts the throttler, suggesting that the throttler is not capable of reassembling fragmented TLS records.

6.3 Domains Targeted

In order to understand if other domains are being targeted, we test the Alexa top 100k domains [2] by replacing them in the TLS SNI field and see if the resulting sessions are throttled.

In the Alexa Top 100k, we find that only `t.co` ***and*** `twitter.com` ***are throttled:*** We also find nearly 600 domains are outright blocked, which suggests that blocking is still the primary means of censorship in Russia.

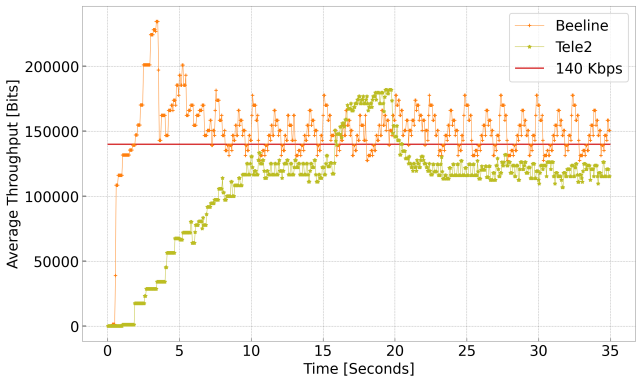

Figure 6: Throughput graphs on Beeline and Tele2 displaying different throttling mechanisms

Focusing on all Twitter-affiliated domains, we test many permutations of the domains known to be throttled, by adding periods before and after the domains, and adding random prefixes/suffixes to them. This step highlights details about string matching policy used by the throttler when inspecting the SNI.

The throttling affects more domains than acknowledged by Roskomnadzor: Early implementation had a loose string matching policy that was corrected after reports suggested that `reddit.com` and `microsoft.com` were also throttled [34]. While this was fixed for `t.co`, we find that a more relaxed string matching is still in effect for other domain strings. Specifically, domain strings such as `*.twimg.com` and `*twitter.com` (e.g. `throttletwitter.com`) continued to be throttled. However, according to our later measurements on April 2, `*twitter.com` is no longer throttled except for the exact matches (e.g. `www.twitter.com`, `api.twitter.com`). It is also worth noting that in an official statement, Roskomnadzor claimed that the throttling is only being applied to the "delivery of audio, video content, and graphics", and that other Twitter functionalities are "delivered without restrictions" [43]. However, we find that among the throttled domains is `abs.twimg.com`, which hosts large Javascript files essential for Twitter to function.

6.4 TTL Measurement

To identify where in the network path throttling occurs, we employ a TTL-based technique similar to traceroute. The IP Time To Live (TTL) field controls how many network hops a packet can traverse. In our measurement, each throttled vantage point establishes a TCP connection with our university's server. Then, using `nfqueue` [29], we insert a Client Hello packet containing a triggering SNI with increasing TTL values and attempt some data transfer. If we identify some TTL value N where we do not observe throttling but TTL $N+1$ results in throttling, then we infer that the throttler operates between the N and $N+1$ hops. This technique allows us to estimate the network location of the throttling infrastructure.

The throttling device is located close to end-users: For all seven vantage points our test shows that the throttling devices operate within the first five hops. Furthermore, in Beeline and Ufanet cases, ICMP TTL-exceeded messages were returned from routable IP addresses. We checked those IP addresses using BGP prefix and ASN lookup, and we found hops both before and after throttling

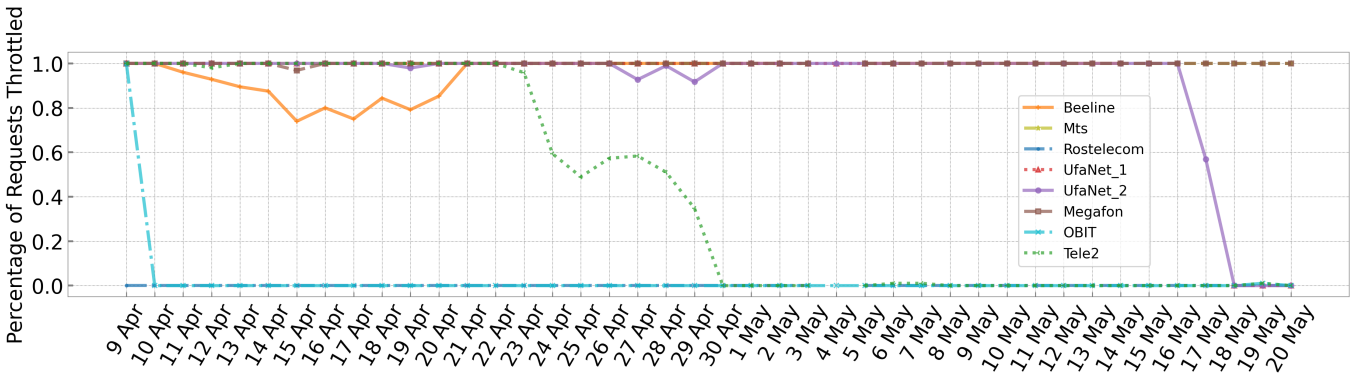

Figure 7: Longitudinal percentage of requests throttled on vantage points

occurred were located inside the clients' ISP network. This result is consistent with a letter sent out by Roskomnadzor to ISPs, where it indicates that the TSPU devices should ideally be installed before carrier-grade NAT devices [20, 41]. Since the installation is close to end-users, as opposed to being at country's border link, domestic traffic is also subject to inspection and censorship from TSPU devices. For example, we confirm that a connection with a Twitter SNI between two Russian hosts is throttled in the same way as if it were a cross-border connection.

We use a similar technique to locate ISPs' blocking devices as well. In this case, we send crafted HTTP requests containing known censored domains iteratively with increasing TTL values, which would trigger blocking devices to return an ISP's "blockpage". In networks where we can estimate the location of the blocking devices, we find that they were between hops 5-8. As this differs from our results for the throttling devices, it suggests that they are not co-located and may be separately managed. Furthermore, in some networks, we also find the throttling devices performing reset-based blocking: on our Megafon vantage point, we observe that once a triggering HTTP request passed hop 2 (the hop after which throttling occurs), a TCP RST terminates the connection. In addition, once the triggering request passes hop 4, the ISP's blockpage is returned. This suggests that the devices performing throttling are also capable of blocking, and that they likely operate independent of the ISP-controlled blocking devices.

6.5 Symmetry of Throttling

We investigate if the throttling is symmetric, i.e. does the throttling equally affect traffic originating from Russia as well as traffic coming into Russia? To do this, we modify Quack Echo, a remote measurement tool that leverages echo servers within a censoring country to measure censorship from outside the country [52]. Briefly, Quack Echo works by sending packets that are specifically crafted to trigger DPI policies to servers running the echo protocol, which, upon receiving the data, will reflect the data back to the sender. We discover 1,297 Russian servers running the echo protocol on port 7, and use them in our Quack Echo measurements. We did not observe any throttling when we connected to these echo servers and sent triggering Client Hellos (which the servers echoed back).

We followed up with our in-country vantage points, as we previously observed that even if the server sent a triggering Client Hello, the connection was throttled. We discover that if the TCP connection is initiated outside Russia to a server inside, we could not trigger throttling. Throttling is triggered (by a Client Hello in either direction) only if a local (in-ISP) client starts a TCP connection with an outside server.

From this we conclude ***that throttling is not symmetric and can only be triggered by connections initiated locally from within Russia:*** This asymmetric nature of the throttling makes it challenging for researchers to study from outside using existing remote measurement tools [38, 48, 52].

6.6 Throttler's State Management

Since throttling usually requires maintaining state, it is necessarily limited by memory, disk space, CPU, *etc.* We are interested in learning about the policies that are used to determine when to discard a state and stop monitoring the associated TCP session. We specifically investigate whether the throttler discards an active (open, data transfer below throttling rate) or inactive session (open, idle) after a time period, or after observing a FIN or RST from either endpoint.

The throttler maintains state for ≈10 minutes for inactive sessions: For an open but inactive (no packets transferred) TCP session, we find that after ≈10 minutes of inactivity, we do not observe throttling. This 10-minute value corresponds to the result we observe in most experiments and is not necessarily a precise threshold of the throttler's state management, which may depend on a variety of factors such as the throttler's operational load, size of the network, *etc.* For active sessions, however, we still observe throttling even two hours into the experiments. This suggests that the threshold for active sessions may be much larger than for inactive sessions, an observation consistent with previous studies [24]. Previous work also found that sending FIN-ACK or RST-ACK could force some middleboxes into discarding a session's state [24, 54]. However, based on our experiments, we found no evidence of the throttler suspending monitoring after seeing a FIN or RST packet from either endpoint.

6.7 Longitudinal Analysis

The throttling is sporadic and inconsistent over time: From our longitudinal measurements, we observe that the throttling occurs sporadically on some vantage points, suggesting that the system is still under active testing and development. For instance, on March 19, we notice that throttling was lifted on our OBIT vantage point for about two days. This correlates with an article about OBIT's service outages which also reported that OBIT had to exclude the TSPU devices from the routing path to restore operation [18].

We also find that on some vantage points the throttling is stochastic in nature, depending on possible routing changes and load balancing. Figure 7 illustrates our findings. Note that OBIT and Tele2 lifted the throttling much earlier before the official announcement on May 17, after which all other landline networks also ceased to throttle Twitter.

7 CIRCUMVENTION

We find and verify several circumvention strategies that can bypass the throttling, based on the insights we obtain by reverse engineering the throttler. Prepending Client Hello records with other TLS records, such as Change Cipher Specs, which is semantically valid, allows us to bypass the throttling (refer to § 6.2). Another strategy is splitting sensitive Client Hellos into multiple TCP packets, either by decreasing Window Size [19, 58] or by inflating the packet with padding extension [25] (refer to § 6.2). Moreover, we can leverage the fact that the throttler discards inactive and unrecognized sessions, by keeping connections idle for around ten minutes or prepending a fake, random packet with lower TTL of size more than 100 bytes (refer to § 6.2, 6.6). Finally, encrypted proxies or VPNs can bypass the throttling, as expected.

While these strategies are effective, only power users are likely aware and capable of adopting them. Therefore, we recommend browsers and websites implement efforts to encrypt the SNI such as using TLS Encrypted Client Hello (ECH), to make it more difficult for censors to throttle based on SNI.

8 DISCUSSION AND CONCLUSION

The throttling of Twitter in Russia marks the first acknowledged use of targeted throttling as a means to put pressure on social media sites. Twitter's compliance [45] proves the success of throttling as a censorship technique. At the time of submission of this paper, Russia already moved to pressuring Google to remove anti-government content from YouTube [40], using the threat of throttling.

The emerging censorship technique of throttling sets a dangerous precedent for all countries that seek to discourage citizens from accessing prohibited resources. The proliferation of "dual-use" technologies such as DPI devices has equipped censors around the world with a more complex toolkit to implement more advanced techniques than outright blocking. From the censor's perspective, the noisy nature of throttling makes it effective and economical to implement, but challenging for users to attribute and difficult for researchers to measure, especially since current censorship detection platforms [33, 35, 50] that focus on blocking are not yet equipped to monitor throttling. We hope that our findings serve as a wake-up call to censorship researchers and encourage future work in detecting and circumventing this emergent censorship technique.

REFERENCES

[1] Alexander Khinshtein: Verification of users on the Internet is a matter of time, 2021. https://tass.ru/interviews/11032409.
[2] Alexa top 1,000,000 sites, 2019. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.
[3] Alice, Bob, Carol, J. Beznazwy, and A. Houmansadr. How china detects and blocks shadowsocks. In *Proceedings of the ACM Internet Measurement Conference*. Association for Computing Machinery.
[4] C. Anderson. Dimming the internet: Detecting throttling as a mechanism of censorship in iran, 2013.
[5] L. E. Andrey Viktorov. Russia twitter throttle dataset, 03 2021. https://github.com/4ndv/russia-twitter-throttle.
[6] Anonymous. The collateral damage of internet censorship by dns injection. *SIGCOMM Comput. Commun. Rev.*, 2012.
[7] K. Bock, Y. Fax, K. Reese, J. Singh, and D. Levin. Detecting and evading censorship-in-depth: A case study of iran's protocol whitelister. In *Workshop on Free and Open Communications on the Internet*. USENIX Association.
[8] A. Chaabane, T. Chen, M. Cunche, E. De Cristofaro, A. Friedman, and M. A. Kaafar. Censorship in the wild: Analyzing internet filtering in syria. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. Association for Computing Machinery, 2014.
[9] Comparing traffic policing and traffic shaping for bandwidth limiting, 2019. https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.html.
[10] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, and S. Saroiu. Glasnost: Enabling end users to detect traffic differentiation. In *USENIX Symposium on Networked Systems Design and Implementation*. USENIX Association, Apr. 2010.
[11] R. Ensafi, D. Fifield, P. Winter, N. Feamster, N. Weaver, and V. Paxson. Examining how the great firewall discovers hidden circumvention servers. In *Proceedings of the ACM Internet Measurement Conference*. Association for Computing Machinery, 2015.
[12] R. Ensafi, J. Knockel, G. Alexander, and J. R. Crandall. Detecting intentional packet drops on the internet via tcp/ip side channels. In *International Conference on Passive and Active Network Measurement*. Springer, 2014.
[13] R. Ensafi, P. Winter, A. Mueen, and J. Crandall. Analyzing the great firewall of china over space and time. *Proceedings on Privacy Enhancing Technologies*, 2015.
[14] G. Esfandiari. Iran admits throttling internet to 'preserve calm' during election, 2013. https://www.rferl.org/a/iran-internet-disruptions-election/25028696.html.
[15] L. Evdokimov. Iran protests: Dpi blocking of instagram, 2018. https://ooni.org/post/2018-iran-protests-pt2/.
[16] O. Farnan, A. Darer, and J. Wright. Poisoning the well: Exploring the great firewall's poisoned dns responses. In *Proceedings of the ACM on Workshop on Privacy in the Electronic Society*. Association for Computing Machinery, 2016.
[17] T. Flach, P. Papageorge, A. Terzis, L. Pedrosa, Y. Cheng, T. Karim, E. Katz-Bassett, and R. Govindan. An internet-wide analysis of traffic policing. In *Proceedings of the SIGCOMM Conference*. Association for Computing Machinery, 2016.
[18] A. Gavrilyuk. Failures for independence - traffic got lost in the equipment of "sovereign runet", 04 2021. https://www.kommersant.ru/doc/4763212.
[19] GoodbyeDPI GitHub Repository, 2021. https://github.com/ValdikSS/GoodbyeDPI.
[20] hsto.org, 2019. https://hsto.org/webt/wk/tk/ud/wktkudgaf5uslgn-gzuj58p-xae.png.
[21] A. International. Iran: Internet deliberately shut down during november 2019 killings – new investigation, 2020. https://www.amnesty.org/en/latest/news/2020/11/iran-internet-deliberately-shut-down-during-november-2019-killings-new-investigation/.
[22] A. Januta and M. Funakoshi. Myanmar's internet suppression, 2021. https://graphics.reuters.com/MYANMAR-POLITICS/INTERNET-RESTRICTION/rlgpdbreepo/.
[23] A. Kakhki, A. Razaghpanah, A. Li, H. Koo, R. Golani, D. Choffnes, P. Gill, and A. Mislove. Identifying traffic differentiation in mobile networks. In *Proceedings of the 2015 Internet Measurement Conference*. Association for Computing Machinery, 2015.
[24] S. Khattak, M. Javed, P. D. Anderson, and V. Paxson. Towards illuminating a censorship monitor's model to facilitate evasion. In *Workshop on Free and Open Communications on the Internet*. USENIX Association, 2013.
[25] A. Langley. A Transport Layer Security (TLS) ClientHello Padding Extension. RFC 7685.
[26] F. Li, A. M. Kakhki, D. Choffnes, P. Gill, and A. Mislove. Classifiers unclassified: An efficient approach to revealing ip traffic classification rules. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, page 239–245, New York, NY, USA, 2016. Association for Computing Machinery.
[27] F. Li, A. A. Niaki, D. Choffnes, P. Gill, and A. Mislove. A large-scale analysis of deployed traffic differentiation practices. In *Proceedings of the ACM Special Interest Group on Data Communication*. Association for Computing Machinery, 2019.

[28] F. Li, A. Razaghpanah, A. M. Kakhki, A. A. Niaki, D. Choffnes, P. Gill, and A. Mislove. Lib•erate, (n): A library for exposing (traffic-classification) rules and avoiding them efficiently. In *Proceedings of the 2017 Internet Measurement Conference*, IMC '17, page 128–141, New York, NY, USA, 2017. Association for Computing Machinery.
[29] libnetfilter queue documentation, 2000. https://netfilter.org/projects/libnetfilter_queue/doxygen/html.
[30] G. Lowe, P. Winters, and M. L. Marcus. The great DNS wall of china. In *Technical Report*. New York University, 2007.
[31] Z. Media. In st. petersburg, participants of the torchlight procession were detained, who carried flags with the roskomnadzor logo, 2021. https://zona.media/news/2021/03/30/unleash-twitter.
[32] menlo. The menlo report, 2012. https://www.dhs.gov/sites/default/files/publications/CSD-MenloPrinciplesCORE-20120803_1.pdf.
[33] A. A. Niaki, S. Cho, Z. Weinberg, N. P. Hoang, A. Razaghpanah, N. Christin, and P. Gill. ICLab: A global, longitudinal internet censorship measurement platform. In *Symposium on Security & Privacy*. IEEE, 2020.
[34] Slowdown of twitter in russia, 2021. https://ntc.party/t/twitter/907.
[35] Open Observatory of Network Interference (OONI). OONI Website. https://ooni.org/, 2021.
[36] J. Park and J. Crandall. Empirical study of a national-scale distributed intrusion detection system: Backbone-level filtering of html responses in china. IEEE, 2010.
[37] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson. Global measurement of DNS manipulation. In *USENIX Security Symposium*. USENIX Association, 2017.
[38] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson. Global measurement of DNS manipulation. In *USENIX Security Symposium*. USENIX Association, 2017.
[39] R. Ramesh, R. S. Raman, M. Bernhard, V. Ongkowijaya, L. Evdokimov, A. Edmundson, S. Sprecher, M. Ikram, and R. Ensafi. Decentralized control: A case study of Russia. In *Network and Distributed System Security*. The Internet Society, 2020.
[40] Reuters. Russia gives google 24 hours to delete banned content, 2021. https://www.reuters.com/technology/russia-gives-google-one-day-delete-banned-content-threatens-slowdown-2021-05-24/.
[41] Report about Roskomnadzor's letter to ISPs, 2019. https://habr.com/ru/post/459894/.
[42] Roskomnadzor takes measures to protect russian citizens from the influence of illegal content, 2021. https://rkn.gov.ru/news/rsoc/news73464.htm.
[43] Twitter is slowed down normally, 2021. https://rkn.gov.ru/news/rsoc/news73480.htm.
[44] Roskomnadzor announces its decision to extend measures to slow down Twitter traffic until May 15 this year, 2021. https://rkn.gov.ru/news/rsoc/news73536.htm.
[45] Twitter informed Roskomnadzor of the progress in removing prohibited materials, 2021. https://rkn.gov.ru/news/rsoc/news73620.htm.
[46] On the partial removal of measures to slow down twitter traffic, 2021. https://rkn.gov.ru/news/rsoc/news73632.htm.
[47] Russia says Twitter complying with demand to remove 'banned content', 2021. https://www.reuters.com/technology/russia-says-twitter-is-complying-with-demand-remove-banned-content-2021-04-30/.
[48] W. Scott, T. Anderson, T. Kohno, and A. Krishnamurthy. Satellite: Joint analysis of CDNs and network-level interference. In *USENIX Annual Technical Conference*. USENIX Association, 2016.
[49] J. Sherman. Kashmir internet shutdown continues, despite supreme court ruling, 2020. https://thediplomat.com/2020/08/kashmir-internet-shutdown-continues-despite-supreme-court-ruling/.
[50] R. Sundara Raman, P. Shenoy, K. Kohls, and R. Ensafi. Censored Planet: An Internet-wide, Longitudinal Censorship Observatory. In *ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 2020.
[51] Telegram, 2021. https://t.me/roskomsvoboda/6619.
[52] B. VanderSloot, A. McDonald, W. Scott, J. A. Halderman, and R. Ensafi. Quack: Scalable remote measurement of application-layer censorship. In *USENIX Security Symposium*. USENIX Association, 2018.
[53] A. Viktorov. Is my Twitter slow or what?, 2021. https://lynx.pink/is-my-twitter-slow-or-what/.
[54] Z. Wang, S. Zhu, Y. Cao, Z. Qian, C. Song, S. V. Krishnamurthy, K. S. Chan, and T. D. Braun. SymTCP: Eluding stateful deep packet inspection with automated discrepancy discovery. In *Network and Distributed System Security*. The Internet Society, 2020.
[55] A. Wilhelm. The internet in iran is crawling, conveniently, right before planned protests, 2010. https://thenextweb.com/news/internet-iran-crawling-conveniently-planned-protests.
[56] P. Winter and S. Lindskog. How the great firewall of china is blocking tor. In *Workshop on Free and Open Communications on the Internet*. USENIX Association, 2012.
[57] X. Xu, Z. M. Mao, and J. A. Halderman. Internet censorship in china: Where does the filtering occur? In N. Spring and G. F. Riley, editors, *Passive and Active Measurement*. Springer Berlin Heidelberg, 2011.
[58] zapret v.39, 2021. https://github.com/bol-van/zapret.
[59] Y. Zhang, Z. Mao, and M. Zhang. Detecting traffic differentiation in backbone isps with netpolice. In *Proceedings of the SIGCOMM Conference on Internet Measurement*. Association for Computing Machinery, 2009.

A APPENDIX

A.1 Timeline of the event

- Starting from March 10, 2021, Russia has started throttling access to Twitter-related services. At 10:30 AM local time, Roskomnadzor issued an official explanation for the throttling, saying that the government had "taken measures to protect Russian citizens from the influence of unlawful content", alluding to Twitter's non-compliance with Russian content takedown requests [42]. According to the statement, the slowdown was be implemented on 100% of mobile services and 50% of landline services. On the same day, internet user ValdikSS disclosed that the relaxed regular expression matching rule `*t.co*` was being used, which caused considerable collateral damage [34] to non-Twitter sites such as microsof**t.co** and redd**it.co**m.
- On March 11, more than 24 hours after the onset of the event, the `*t.co*` matching rule was apparently patched and only an exact match of `t.co` can trigger throttling. Shortly after the patch, Roskomnadzor issued a press release that stated "Twitter is throttled as expected" [43]. Later that day, we began collaboration with local activists, obtained access to local vantage points and started running measurements to study the event.
- On March 30, Russian police detained four members of the Vesna movement who were carrying flags with the Roskomnadzor logo to protest the throttling of Twitter [31].
- On April 2, the regex matching rule `*twitter.com` was restricted to `twitter.com`, possibly in response to our initial report. On the same day, a statement from Roskomnadzor disclosed that Twitter was fined 8.9 million rubles for failure to respond to content takedown requests.
- On April 5, following a talk with Twitter that took place on April 1st, Roskomnadzor issued another statement acknowledging that Twitter has "increased the speed of removing prohibited materials" [44]. However, since the content removal speed was still not aligned with Russia's law, the throttling was extended to May 15th.
- On April 28, Roskomnadzor said Twitter was "complying with demand to remove banned content" and that it and Twitter agreed to establish a direct line of communication between the watchdog and Twitter's moderation service [47].
- On May 14, Twitter informed Roskomnadzor that it had fulfilled the requirements to remove content prohibited in Russia and requested the throttling to be lifted [45].
- Our continual measurement indicates that the throttling was lifted for landlines on May 17 around 16:40 Moscow time. On 17:00, Roskomnadzor issued an official statement in which it "appreciates the efforts of Twitter to comply with the requirements of Russian legislation" and therefore removed throttling of Twitter on landlines while continued throttling Twitter on mobile operators [46].

- On May 24, Roskomnadzor pressured Google to remove anti-government content from YouTube [40] within 24 hours, using the same threat of throttling Google's traffic inside the country.